Hybrid Particle-Field Molecular Dynamics Simulations: Parallelization and Benchmarks

Ying Zhao,[a]*,† Antonio De Nicola,[a,b] Toshihiro Kawakatsu,[c] and Giuseppe Milano*[a,b]

The parallel implementation of a recently developed hybrid scheme for molecular dynamics (MD) simulations (Milano and Kawakatsu, J Chem Phys 2009, 130, 214106) where self-consistent field theory (SCF) and particle models are combined is described. Because of the peculiar formulation of the hybrid method, considering single particles interacting with density fields, the most computationally expensive part of the hybrid particle-field MD simulation can be efficiently parallelized using a straightforward particle decomposition algorithm. Benchmarks of simulations, including comparisons of serial MD and MD-SCF program profiles, serial MD-SCF and parallel MD-SCF program profiles, and parallel benchmarks compared with efficient MD program GROMACS 4.5.4 are tested and reported. The results of benchmarks indicate that the proposed parallelization scheme is very efficient and opens the way to molecular simulations of large scale systems with reasonable computational costs. © 2012 Wiley Periodicals, Inc.

DOI: 10.1002/jcc.22883

Introduction

Simulation methods can be classified on the basis of the nature of simulated objects. In particular, in the case of soft matter simulations, particle-based models such as full atomistic or specific coarse-grained models and the corresponding simulation techniques are generally described as molecular simulations.[1–3] Because of their possibility of straightforward treatments of chemical details, molecular simulations are very useful to characterize structural and dynamical properties of polymeric materials[4,5] and biomolecules.[6] In the framework of this class of simulation methods, molecular dynamics (MD) method is a widely applied computational technique.[7–10]

A different class of methods is the one based on field representation. The framework in which this approach is developed is classical density functional theory (DFT). This theoretical framework has been extensively used to study inhomogeneous complex fluids.[11,12] In such a field-based approach, the model systems are not represented by particles but by density fields and their behavior on larger time and length scales are studied. Particularly popular is the self-consistent field (SCF) theory, where the mutual interactions between segments are decoupled and replaced by static external fields.[11] In the SCF theory, external fields depend on the statistical averages of the spatially inhomogeneous density distributions of particles created by the independent molecules interacting only with these external fields. Such external fields and the particle density distributions have to be determined self-consistently. Numerous applications of block copolymers,[13–17] proteins,[18] polymer composites,[19] and colloidal particles[20,21] have demonstrated that the SCF theory is a useful and powerful method. The main disadvantage of this class of methods is the limitation to simple and generic coarse grain models, due to the difficulty in including chemical details into the model.

Müller and Smith[22] introduced a hybrid approach in the framework of SCF theory by combining it with a Monte Carlo (MC) simulation of polymer chains to study phase separation in binary polymer mixtures. This approach has been widely and successfully applied by Müller et al, and Daoulas et al, to coarse-grained models of diblock copolymer thin films[22,23] and polymer nanocomposites.[24,25] Sides et al.[26] successfully developed a similar hybrid particle-field method based on the use of "cavity" functions to exclude the fluid components from the interior of solid particles, which explicitly retain the particle coordinates as degrees of freedom unlike SCF-DFT.[27–29]

One of the advantages of this hybrid approach is the lack of any limitations in treating complex molecular architectures and/or intramolecular interactions. Very recently, a hybrid particle field approach, where the MD method based on particle description is combined with field description, was proposed and the implementation suitable for the treatment of atomistic force fields and/or specific coarse-grained models has been reported.[30,31] The idea behind this combined method is to obtain a strategy, as far will be possible, having the main

[a] *Y. Zhao, A. De Nicola, G. Milano*
Dipartimento di Chimica e Biologia and NANOMATES, Research Centre for NANOMAterials and nanoTEchnology at Università di Salerno, I-84084 via Ponte don Melillo Fisciano (SA), Italy
E-mail: yzhao@unisa.it or gmilano@unisa.it

[b] *A. De Nicola, G. Milano*
IMAST Scarl-Technological District in Polymer and Composite Engineering, P.le Fermi 1, 80055 Portici (NA), Italy

[c] *T. Kawakatsu*
Department of Physics, Tohoku University, Aoba, Aramaki, Aoba-ku, Sendai 980-8578, Japan

†*Present address:* State Key Lab of Molecular Reaction Dynamics, Dalian Institute of Chemical Physics, Dalian, Liaoning 116023, China. E-mail: zhaoying@dicp.ac.cn.

Contract/grant sponsor: MIUR; contract/grant number: PRIN2008
Contract/grant sponsors: FIRB Rete ITALNANONET, Grant-in-Aid for Science from the Ministry of Education, Culture, Sports, Science and Technology, Japan.

advantages and avoiding the main disadvantages of both SCF and atomistic MD simulation techniques.

Hybrid MD-SCF simulations are computationally more convenient and efficient than the corresponding full atomistic MD simulations. The main reason of this efficiency is the way in which the multibody problem of MD simulation is solved. In the hybrid simulation method, the most expensive part of a MD simulation, i.e., the calculation of intermolecular nonbonded forces, is replaced by the calculation of interaction forces between independent particles in an average density field.

A further way to achieve larger length and time scales is the exploitation of techniques for parallelization. According to the features of parallel architectures, there are two models: shared memory through OpenMP[32] and distributed memory through message passing interface (MPI)[33] extensively applied for parallelizing MD algorithms.[34–46] Many different MD software packages such as GROMACS,[34] DL_POLY,[35] IMD,[36] M.DynaMix,[37] YASP,[44] NAMD,[45] LAMMPS,[46] successfully and widely applied parallelized algorithms. Furthermore, the implementation of all-atom MD programs on graphical processor units (GPU)[47] encourages the development of parallelization schemes for molecular simulations.

In the framework of the model of distributed memory through MPI, several strategies of parallelization of a MD simulation program have been reported. The parallelization strategies can be divided in three main schemes including particle-(atom)-decomposition also called replicated-data algorithm, domain-(spatial)-decomposition algorithm, and force-decomposition algorithm. References [48–53] include good overviews of these various techniques. The most time-consuming part of the MD simulation is the computation of the nonbonded force acting on the particles. So a procedure in which a predetermined set of force calculations is assigned to each processor is an effective choice. The simplest way to afford this is to distribute a subgroup of particles into each processor and the chosen distribution is fixed for the duration of the simulation, which is called particle-(atom)-decomposition algorithm.

To our best knowledge, although hybrid particle-field approaches are very promising, parallelization schemes, benchmarks on the parallelization of this type of simulations have not been reported until now. Therefore, the aim of this study is to propose a simple and effective parallelization strategy based on particle decomposition algorithm for distributed–memory machines using MPI suitable for hybrid particle-field MD simulations. Especially, results and benchmarks of simulations, including comparisons of serial MD and MD-SCF program profiles, serial MD-SCF and parallel MD-SCF program profiles, and parallel benchmarks compared with efficient MD program GROMACS 4.5.4 are tested and reported in a more detail.

Hybrid particle field MDs

General scheme In this section, a brief exposition of the recently developed hybrid particle-field MD simulation scheme is described. This section is intended to quickly guide the reader to get the basis of the algorithms of the methodology and understand the framework of the present investigation. To obtain this approach in more detail, the readers should refer to reference [30] where the complete derivation and the implementation are described and to reference [11] for a general review of SCF methods.

The main feature of the hybrid particle-field approach, is that the evaluation of the nonbonded force and its potential between atoms of different molecules, i.e., the most computationally expensive part of MD simulations, is replaced by an evaluation of the external potential that depends on the local density at position $\boldsymbol{r}$ obtained in SCF simulations. According to the spirit of SCF theory, a many body problem like molecular motion in systems composed of many molecules is reduced into the problem of deriving the partition function of a single molecule in an external potential $V(\mathbf{r})$. Then nonbonded force between atoms of different molecules can be obtained from a suitable expression of the $V(\mathbf{r})$ and its derivatives.

In the framework of the SCF theory, a molecule is regarded to be interacting with the surrounding molecules not directly but through a mean field. On the basis of this picture, the Hamiltonian of a system that is composed of M molecules can be split into two parts as $\hat{H}(\Gamma) = \hat{H}_0(\Gamma) + \hat{W}(\Gamma)$, where Γ specifies a point in the phase space that is used as shorthand for a set of positions and momentums of all atoms in the system. Hereafter, the symbol ^ (hat) indicates that the associated physical quantity is a function of the microscopic states described by the phase space Γ.

Assuming the canonical (*NVT*) ensemble, the partition function of this system is given by:

$$Z = \frac{1}{M!}\int d\Gamma \exp\{-\beta[\hat{H}_0(\Gamma) + \hat{W}(\Gamma)]\}, \quad (1)$$

where $\hat{H}_0(\Gamma)$ is the Hamiltonian of a reference ideal system composed of M noninteracting chains but with all the intramolecular interaction terms (bond, angle, and nonbonded interactions) that are taken into account in the standard MD simulations. On the other hand, the deviation from the reference system on account of the intermolecular nonbonded interactions is explained by the term $\hat{W}(\Gamma)$ in Eq. (1).

From microscopic point of view, the density distribution of atoms can be defined as a sum of delta functions centered at the center of mass of each particle as:

$$\hat{\phi}(\mathbf{r};\Gamma) = \sum_{p=1}^{M}\sum_{i=0}^{S(p)}\delta\left(\mathbf{r} - \mathbf{r}_i^{(p)}\right), \quad (2)$$

where $S(p)$ is the number of particles contained in p-th molecule, and $\mathbf{r}_i^{(p)}$ is the positions of particles in p-th molecule. The deviation $\hat{W}(\Gamma)$, according to Eq. (1), from the reference state $\hat{H}_0$ originates from the interactions between molecules. Several assumptions are introduced to calculate this interaction term $\hat{W}(\Gamma)$. First of all, we assume that $\hat{W}(\Gamma)$ depends on Γ only through the particle density $\hat{\phi}(\mathbf{r};\Gamma)$ as:

$$\hat{W}(\Gamma) = W\left(\hat{\phi}(\mathbf{r};\Gamma)\right). \quad (3)$$

Using δ-functional, the partition function in eq. (1) can be rewritten as:

$$Z = \frac{1}{M!}\int D\{\phi(\mathbf{r})\}\int D\{w(\mathbf{r})\} \times \exp\left\{-\beta\left[-\frac{M}{\beta}\ln z + W[\phi(\mathbf{r})] - \int V(\mathbf{r})\phi(\mathbf{r})d\mathbf{r}\right]\right\}, \quad (4)$$

where z is the single molecule partition function, $\phi(\mathbf{r})$ is the average of $\hat{\phi}(\mathrm{r})$ introduced through $\delta[\hat{\phi}(\mathrm{r}) - \phi(\mathrm{r})]$, $w(\mathbf{r})$ is a conjugate field of $\phi(\mathbf{r})$ that appeared in the Fourier representation of the d-functional and $V(\mathbf{r})$ is defined by $V(\mathbf{r}) = w(\mathbf{r})/\mathrm{i}\beta$, respectively.

For evaluating this partition function approximately, the integrals over $\varphi(\mathbf{r})$ and $w(\mathbf{r})$ in eq. (4) are replaced by a Gaussian integral around the most probable state that minimizes the argument of the exponential function on the right side of Eq. (4) (so-called saddle point approximation).

The minimization conditions in the form of functional derivatives result in:

$$\begin{cases} V(\mathbf{r}) = \frac{\delta W[\phi]}{\delta\phi(\mathbf{r})} \\ = -\frac{M}{\beta z}\frac{\delta z}{\delta V(\mathbf{r})} = \left\langle\hat{\phi}(\mathbf{r};\Gamma)\right\rangle = \phi(\mathbf{r}). \end{cases} \quad (5)$$

In term of eq. (5), it is possible to acquire an expression for a density dependent external potential acting on each segment.

Next, we assume that the interaction term W, i.e. the density dependent interaction potential defined in eq. (3), is in a quadratic form of the densities of each component species specified by the index K and has the following form:

$$W[\{\phi_K(\mathbf{r})\}] = \int d\mathbf{r}\left(\frac{k_BT}{2}\sum_{KK'}\chi_{KK'}\phi_K(\mathbf{r})\phi_{K'}(\mathbf{r}) + \frac{1}{2\kappa}\left(\sum_K\phi_K(\mathbf{r}) - \phi_0\right)^2\right), \quad (6)$$

where the second term of the integrand on the right-hand side is the relaxed incompressibility condition, κ is the compressibility that is assumed to be sufficiently small, and ϕ_0 is the total number density of segments (we assume that volume for all segments are the same). Then, the corresponding mean field potential is given by

$$V_K(\mathbf{r}) = \frac{\delta W[\{\phi_K(\mathbf{r})\}]}{\delta\phi_K(\mathbf{r})} = k_BT\sum_{K'}\chi_{KK'}\phi_{K'}(\mathbf{r}) + \frac{1}{\kappa}\left(\sum_K\phi_K(\mathbf{r}) - \phi_0\right). \quad (7)$$

Taking the case of a mixture of two components A and B as an example, the mean field potential acting on a particle of type A at position $\boldsymbol{r}$ is given by:

$$V_A(\mathbf{r}) = k_BT[\chi_{AA}\phi_A(\mathbf{r}) + \chi_{AB}\phi_B(\mathbf{r})] + \frac{1}{\kappa}(\phi_A(\mathbf{r}) + \phi_B(\mathbf{r}) - \phi_0). \quad (8)$$

Thus, the force acting on the particle A at position $\mathbf{r}$ imposed by the interaction with the density field is

$$F_A(\mathbf{r}) = -\frac{\partial V_A(\mathbf{r})}{\partial\mathbf{r}} = -k_BT\left(\chi_{AA}\frac{\partial\phi_A(\mathbf{r})}{\partial\mathbf{r}} + \chi_{AB}\frac{\partial\phi_B(\mathbf{r})}{\partial\mathbf{r}}\right) - \frac{1}{\kappa}\left(\frac{\partial\phi_A(\mathbf{r})}{\partial\mathbf{r}} + \frac{\partial\phi_B(\mathbf{r})}{\partial\mathbf{r}}\right). \quad (9)$$

Electrostatics effects can be partially included in the hybrid models by a suitable choice of χ parameters. In particular, as already explored in the reference particle–particle models for biological phospholipids,[6] cross interaction terms between hydrophilic and hydrophobic beads can be tuned to correctly reproduce the structure and the phase behavior. In the case of large electrostatic interactions (in systems such polyelectrolytes, charged protein surfaces etc.), the field approach can be extended to charges, including in the density functional terms describing the interaction of charges with charge density fields.[12] In particular, this issue can be solved implementing Ewald method, PPPM method, or directly solving the Poisson equation for the electrostatic potential. This approach will allow also the development of a suitable particle-field model for the simulation of explicit water at atomistic level.

Another important aspect of hybrid particle-field models is the mapping between the timescales of atomistic or coarse-grained particle–particle simulations and the particle-field ones. Usually, due to much smoother potentials, the dynamics in particle-field simulations is faster. An evaluation of a conversion factor between particle–particle and particle-field simulations has been done for lipid bilayer systems comparing the diffusion coefficients.[54] The factor between coarse-grained particle-field lipid and water beads and the corresponding particle–particle models is about four with respect to coarse-grained models and about twenty with respect to atomistic simulations. This factor, of course, depends on the specific coarse-grained model and should be evaluated in the same way for different models.

Implementation of serial MD-SCF program To connect particle and field models, for the proposed hybrid MD-SCF scheme, it is necessary to obtain a smooth coarse-grained density function directly from the particle positions Γ. Let us denote this procedure as

$$\bar{S}\{\hat{\phi}(r;\Gamma)\} = \phi(r), \quad (10)$$

where $\bar{S}$ is a symbolic name of the mapping from the particle positions to the coarse-grained density. The procedure to obtain the coarse-grained density starting from particle position has been explained and reported in ref. [30].

The iteration algorithm used in MD-SCF approach is outlined in Figure 1. According to the initial configurations of the system (at time t_0), a starting value of the coarse-grained density is obtained. The coarse-grained density is defined on a lattice and the values of the density and density gradients at the particles' positions are calculated by linear interpolation.[30] Then, from the density gradients, forces acting on the particles at position $\boldsymbol{r}$ due to the interaction with the density fields are computed according to eq. (9). The total force acting on the particles will be the sum of the intramolecular forces (bonds,

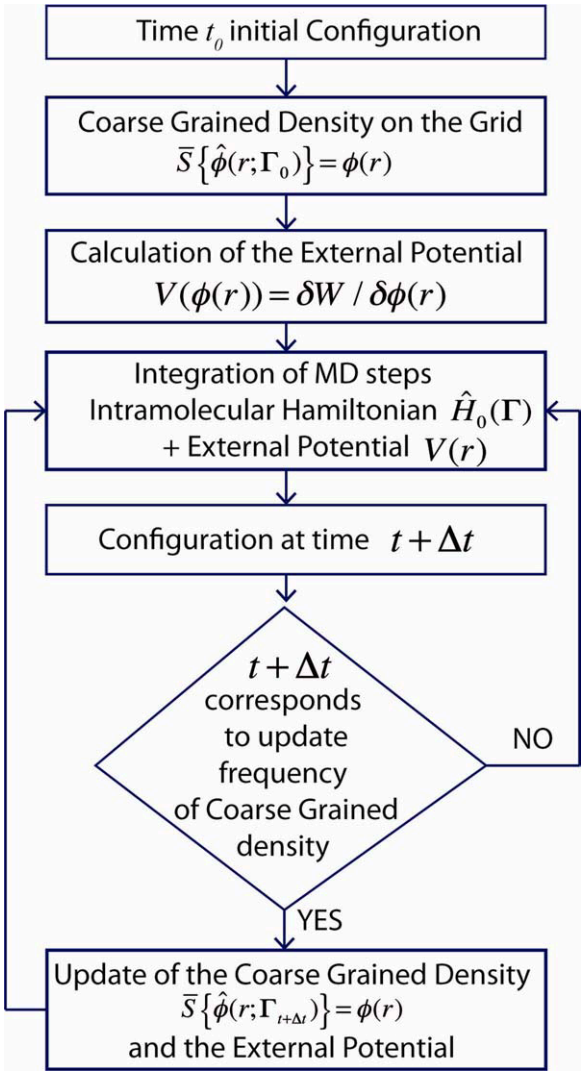

Figure 1. The iteration scheme proposed for hybrid MD-SCF simulations. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

angles, and intramolecular nonbonded forces calculated as in classical MD simulations) and the forces due to the interactions of particles with density fields. After the force calculation, a new configuration will be then obtained by integration of the equation of motion. In principle, at every new configuration an update of the coarse-grained density calculated from the new coordinates should be performed. Test simulations have shown that, due to the collective nature of the density fields, it is possible to define an update frequency of the coarse-grained densities without loss of accuracy.[30,31] In other words, the values of the coarse-grained density at lattice points are not updated at every timestep but only at every prefixed density-update time (Δt_{update}). Then between two updates the values of the densities on the lattice used to interpolate both density and its derivatives will be constant. At every density-update a new coarse-grained density will be obtained and, as outlined in Figure 1, the iteration algorithm converges when the coarse-grained density and the particle-field potential become self-consistent.

The implementation, combining MD with SCF, and the models presented in this article have been successfully validated against reference particle–particle simulations for polymer melts, block copolymers[30,31] and coarse-grained models of phospholipids.[54] It is worth noting that, according to the scheme described above, between two density updates, the integration of the equation of motions is a fully parallelizable problem. In fact, due to the absence of intermolecular force calculations, once that the density field has been updated, for every single molecule, no information is needed from other molecules in terms of particle positions. For the calculation of forces due to the interaction of particles with density fields the only information needed is the density field and its spatial derivatives. These features of the hybrid particle-field formulation, having many noninteracting molecules interacting only with a density field, make this method very suitable to efficiently exploit parallelization.

To obtain a smooth spatial density from particle positions, the simulation box is divided into several cells of size *l*. In particular, particles are assigned, according to their positions, to *n*cell $= n_x{*}n_y{*}n_z$ (where n_x, n_y, and n_z are the numbers of cells in the *x, y,* and *z* directions, respectively). Thus, the total number of mesh points in the lattice where density is calculated is *n*_lattice $= (n_x{+}1){*}(n_y{+}1){*}(n_z{+}1)$.

In the program OCCAM[55] the proposed approach is implemented by looping over the number of particles *N* to calculate the coarse-grained density on the vertexes of the cells where the density field is defined. The computation of density derivatives on the staggered lattice is obtained by looping over the number of lattice points *n*_lattice. As explained above, the calculation of density and its derivatives on the lattice points is performed at every prefixed Δt_{update} according to the positions of the particles in the simulation box. A schematic pseudo code for hybrid MD-SCF simulations is reported in Box 1.

From the point of view of the computational efficiency, hybrid MD-SCF algorithm has a main advantage with respect to classical MD simulations. The most computationally expensive part of the MD simulations, i.e. the evaluation of intermolecular nonbonded force is completely replaced by an evaluation of a particle-field forces originating from the interaction of individual molecules with the density field. This means that at each time step the double loop over particle pairs for the calculation of intermolecular forces of classical MD simulations is replaced by a single loop over *N* particles used to interpolate density gradients at particle position and occasionally, only at the update steps, further two loops are needed. A first loop for the calculation of density on lattice points from particle position (loop over particle number) and a second loop for the calculation of density derivatives (loop over lattice points).

Parallelization scheme

Particle decomposition algorithm. Particle decomposition algorithm involves the distribution of a subgroup of particles into each processor. The chosen distribution can be fixed for the duration of the simulation. In the present work, *N* particles are split evenly among *P* processors and therefore each processor

```
do istep=1,nstep

    Calculation of Intramolecular Forces

 if (istep.eq.update_step.or.eq.1)

   do i=1,n_particles

     Calculation of Coarse-Grained Density

     icell = 1+ (int(x(i)/lx))+ (int(y(i)/ly))*nx + (int(z(i)/lz))*nx*ny
     Distribute fractions of particle i on the vertexes of
     the cell no. icell according to particle position x(i), y(i), z(i)

   enddo

   do i=1,n_lattice

     Calculation of Density Derivatives

   enddo

 endif

  do i=1,n_particles

    Gradients Interpolation(x(i),y(i),z(i))
    Particle Field Forces (Gradx(i), Grady(i), Gradz(i))
    Total Forces = intramolecular+Particle Field Forces

   enddo

    Integration of equation of motion

enddo
```

Box 1. Pseudocode for hybrid MD-SCF simulations.

owns *N/P* particles. It is worth to note that, to reduce the communication between processors, it is convenient to distribute the particles of the same molecules on the same processor. Thus *N* is chosen in a way to avoid that atoms of the same molecule are distributed among two different processors.

At each time step, a processor calculates the forces between its particles and those of the rest of the system. In a classical MD simulation, after updating the positions and momentums of its particles, each processor needs to perform an all-to-all communication to obtain the new positions of all particles in the system for the preparation of the next time step. This makes the efficiency of the classical MD simulations that is parallelized with the use of the particle decomposition scheme much lower than the one obtained with domain decomposition algorithm.

Differently from classical MD simulations, in the formulation of the MD-SCF approach described above, the particle field forces that substitute the expensive calculations of the intermolecular non bonded forces are evaluated between independent molecules and the density fields. This means that it is not necessary to communicate the positions of all the particles and the only communication operation is related to the update of the density fields. As explained before, this operation is not performed at every time step but according to a prefixed frequency. Moreover, the amount of the communication data in one operation is only related to the number of lattice points, which is much less than the number of the positions of all particles. Due to these features of hybrid MD-SCF scheme, we adopted particle decomposition strategy, assigning a given set of molecules to every processor.

The main advantage of particle decomposition algorithm is its simplicity and then its straightforward implementation. Due to an easy load-balance among processors in the calculation of intramolecular forces, including bonds, angles, and torsions, the particle decomposition algorithm can be efficiently used for hybrid MD-SCF simulations. In the following, the particle decomposition algorithm and its implementation for the parallel MD-SCF simulations will be described in more detail.

Implementation of parallel MD-SCF program. The main parts of the calculations involved in a MD-SCF simulation together with their sizes (in terms of loop lengths) are schematized in Figure 2. In particular, in this figure serial and parallel implementations proposed in this article are compared. In the box, a given operation (indicated in blue), and its corresponding loop length of serial implementation (indicated in green) and parallel implementation (indicated in red) are described. Furthermore, communication operation is indicated in yellow. From Figure 2, it is clear that the loops corresponding to the evaluations of both intra molecular and particle-field forces are fully parallelized and no communication operations are needed. It is worth to note that the evaluation of coarse-grained density is also fully parallelized by looping over the number of particles assigned to every processor.

According to the particle decomposition algorithm, at the beginning of simulations *N* particles are assigned to *P* processors. Thus, each processor owns *N/P* particles during the MD-SCF simulations. It is worth to note that all particles of a molecule are assigned to the same processor. Obviously, the amount of calculation of intramolecular interaction forces (bonds, angles, and intramolecular nonbonded interactions) is *Nbonds/P*, *Nangles/P*, and *Nnbpairs/P*, respectively (as shown in Fig. 2). As explained above, in hybrid MD-SCF approach, the computations of intermolecular nonbonded forces of classical MD simulations are replaced by the calculation of forces acting on every particle due to density field. According to eqs. (8) and (9), particle-field interaction potential energy and the relative forces are calculated from particle positions by interpolating the values of density fields and their spatial derivatives. Moreover, the density field (and its derivatives) is not calculated at every time step from particle positions but are updated with a given frequency. This implies that between two updates the calculation of potential energy and forces acting on the particles can be fully parallelized and does not involve any communication among the processors. In this case the amount of force and potential evaluations are *N/P* for every processor.

At the first simulation step and then at every update time Δt_{update}, the density is evaluated on the lattice points. In the proposed implementation, densities are evaluated in each processor for its own *N/P* particles (partial densities). Then, an MPI_ALLREDUCE call with MPI_SUM operator is performed to obtain the total coarse-grained density on the lattice points by summing partial densities owned in every processor. The parallelization scheme for the evaluation of partial density and the communication operation to obtain the total density is shown

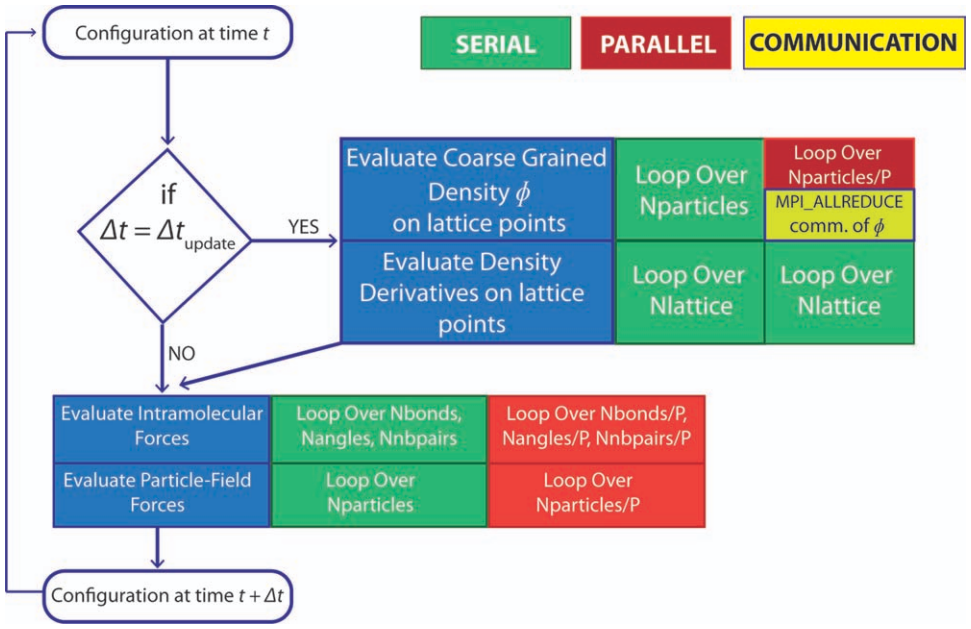

Figure 2. The iteration schemes of serial and parallel MD-SCF programs (OCCAM) are outlined. In this Figure, green parts represent serial implementation in OCCAM and the interaction schemes with red show the parallel implementation. MPI_ALLREDUCE communication part is represented with yellow.

in Figure 3. As an example we consider a system made of eight molecules in a parallel simulation running on four processors. First of all, eight molecules in Figure 3a are divided into four groups, each of which is assigned to one of the four different processors. Thus, each processor owns two molecules (as shown in Fig. 3b). For example, processor P_0 owns molecules 1 and 2, and ϕ_0 indicates the partial density calculated only from the positions of the atoms on these two molecules. Similarly, partial densities ϕ_1 to ϕ_3 related to molecules owned by processors P_1 to P_3 are calculated. Next, each processor needs to expand coarse-grained partial density ϕ_n among all processors and sum ϕ_n to obtain the total coarse-grained density ϕ. This operation corresponds to an all-to-all communication with the total amount of lattice points *n*_lattice. It is worth noting that in typical applications the usual number of particles in a cell is between the order of 1 and 10.[30] This means that usually the size of the coarse-grained density cell can be thirty times lower than the size of the particles positions. We stress that in the proposed implementation the calculation of partial density is fully

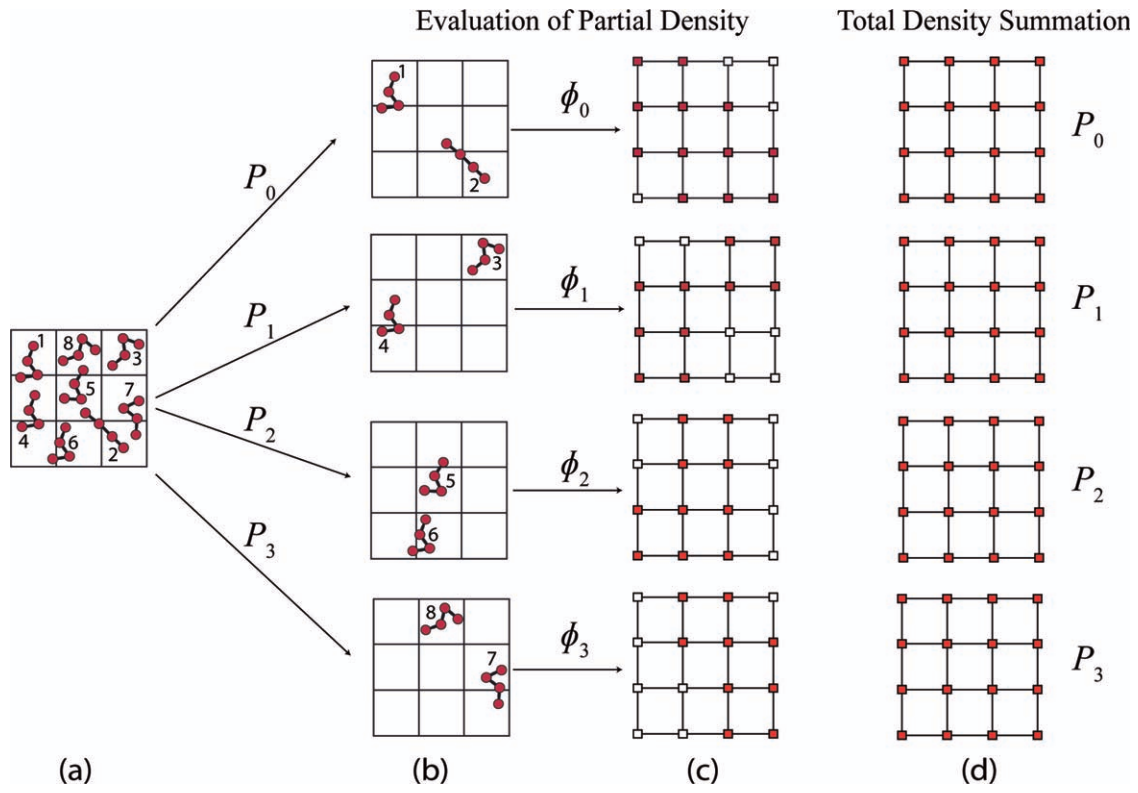

Figure 3. Density-update parallelization shown in an example of eight molecules assigned to four processors. a). There are eight molecules in the three-dimensional box. b). Partition of the eight molecules into four groups assigned to four processors. c). Calculation of the partial density on the vertexes of cells according to the positions of molecules. d). Total density is summed from the partial density of each processor.

Table 1. Descriptions and details of the systems tested to assess efficiency of parallel MD-SCF program on OCCAM.

Models	System name	No. of particles	No. of lattice points	Box size (nm^3)
Monoatomic fluid	MF1	50,000	29 × 29 × 29 (24,389)	20 × 20 × 20
	MF2	100,000	37 × 37 × 37 (50,653)	25.2 × 25.2 × 25.2
	MF3	500,000	34 × 34 × 34 (39,304)	43.09 × 43.09 × 43.09
			62 × 62 × 62 (238,328)	
	MF4	1,000,000	123 × 62 × 62 (472,812)	86.18 × 43.09 × 43.09

	System name	No. of particles	No. of lipids (no. of particles)[a]	No. of water molecules	Cell size l (σ)	No. of lattice points	Box size (nm^3)
Lipids and water	LW1	307,200	4,550 (54,600)	252,600	1.5	59×59×31 (107,911)	40.88 × 40.88 × 20.85
					2.5	36×36×19 (24,624)	
	LW2	1,048,576	53,248 (638,976)	409,600	1.5	187×187×11 (384,659)	130.82 × 130.82 × 6.95
					2.5	112×112×7 (87,808)	

	Chain length[b]	System name	No. of particles	No. of rod chains (no. of particles)[a]	No. of coil chains (no. of particles)[a]	Cell size l (σ)	No. of lattice points	Box size (nm^3)
Rod and coil homopolymer mixtures	30	RC1	30,000	500 (15,000)	500 (15,000)	1.5	22 × 22 × 22 (10,648)	43.36 × 43.36 × 43.46
	100	RC2	600,000	3,000 (300,000)	3,000 (300,000)	1.5	57 × 57 × 57 (185,193)	117.68 × 117.68 × 117.68
						2.5	35 × 35 × 35 (42,875)	
	120	RC3	1,200,000	5,000 (600,000)	5,000 (600,000)	1.5	72 × 72 × 72 (373,248)	148.27 × 148.27 × 148.27
						2.5	43 × 43 × 43 (79,507)	

[a] In parentheses the number of particles are reported. [b] Number of particles per chain.

parallelized and, in contrast to classical MD simulations, particle positions are never communicated between different processors and the communication operation involves only the density field.

Furthermore, to calculate forces, spatial derivatives of the densities are needed. This is the only part that is not parallelized. In principle, the density derivatives could be calculated as the sum of the derivatives of partial densities owned by each processor. However, this means that in the case of a homogeneous system, small values of the derivatives should result in the sum of large positive and negative contributions from each processor. In such a case, an extension of the approach used for the parallel evaluation of the total density derivatives can cause numerical inaccuracy. Thus, we choose to keep this calculation of density derivatives serial. However, every processor has the same amount of the calculation, which makes the parallel implementation balanced and saves the waiting time in every processor.

From the point of view of the computational performances, the simple particle decomposition algorithm implemented in the hybrid particle-field MD-SCF program OCCAM is nearly completely parallelized with communication operations needed only at every density-update time. Thus, the parallel MD-SCF algorithm completely skips large amounts of communication of particles positions which is needed in parallel particle–particle MD program using the same decomposition algorithm.

Programming language, profilers, and machines

The program OCCAM for parallel MD-SCF simulations is written in our group using the FORTRAN 90 programming language, and has been parallelized by using MPI. Standard MPI_BCAST library routine is implemented to send the data from a master task to slave tasks, and MPI_ALLREDUCE with the MPI_SUM operator is used to sum the contributions to total density from the slave tasks each of which has a partial density on the lattice points in a three dimensional box. The tests have been performed on the Cresco1 and Brindisi clusters at the CRESCO ENEA Computer Center (http://www.cresco.enea.it/). Technical details about the used hardware and software are shown in Table S1 (Supporting Information). Furthermore, to compare the serial MD-SCF with full atomistic MD program, the serial profiler of GNU gprof[56] v2.17 are used for the profiling of serial program. On the other hand, parallel profiler TAU[57] v2.19 are used for the profiling of parallel program MD-SCF with 1 cpu (parallel), 32 cpus, and 64 cpus, respectively.

Results and Discussions

To assess the performance efficiency of the parallel MD-SCF program, we carried out simulations of 10,000 time steps for the following three applications: monoatomic fluid (MF), mixtures of lipids and water (LW) as well as mixtures of rod and coil homopolymers (RC). Details of these systems simulated in this study are reported in Table 1. A description of the used models is reported in the Supporting Information.

Comparison of serial MD and MD-SCF program profiles

To compare the efficiencies of serial MD and MD-SCF programs, the system RC1 including 30,000 particles has been simulated for 1,000,000 time steps. In the serial MD-SCF simulation the density-update using particle positions has been done every 300 steps and the total number of lattice points

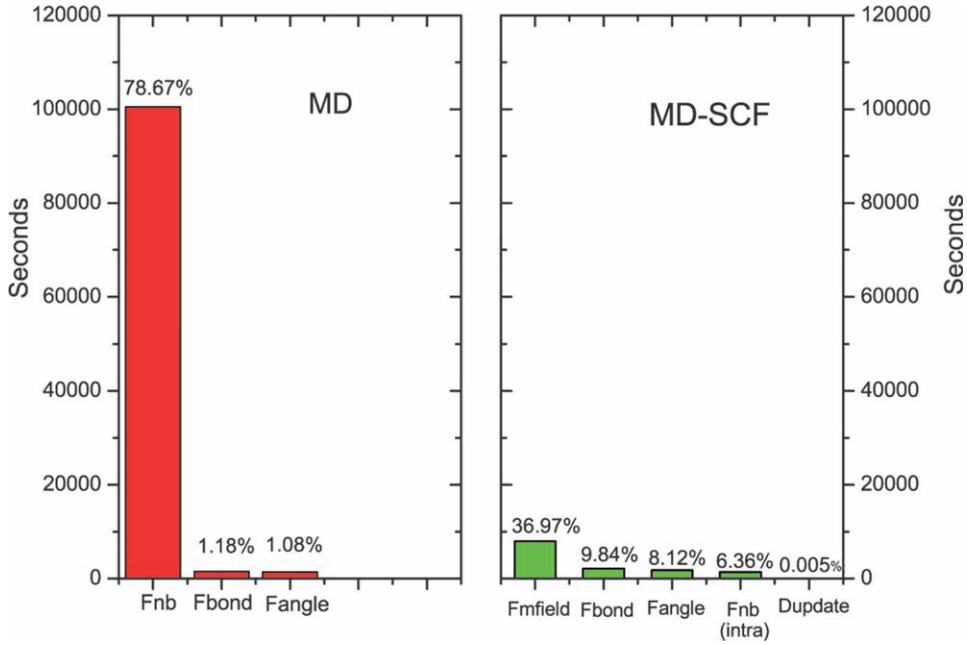

Figure 4. Comparison of serial MD and MD-SCF profiles using gprof v2.17 profiler. The system RC1 including 30,000 particles has been simulated for 1,000,000 time steps with update frequency $\Delta t_{update} = 300$ time steps. The computational times (in second) corresponding to nonbonded, particle-field, bond, and angle forces are shown.

where the coarse-grained density is defined is $22 \times 22 \times 22 = 10{,}648$, which corresponds to the cell size $l = 1.5\sigma$ (σ is the diameter of one particle, $\sigma = 1.4$ nm).

The CPU times spent in classical serial MD and MD-SCF simulations has been compared. The partial simulation time of the main parts and their percentages in the total simulation time are obtained from the profiler gprof v2.17 and compared in Figure 4. In classical serial MD program, F_{nb} indicates the time spent on the calculation of nonbonded forces, including the update of the neighbor list at every 10 time steps. However, in serial MD-SCF simulations, the neighbor list is updated with the same frequency but involves only intramolecular interactions. Thus, F_{nb} corresponds to the time spent to calculate intramolecular nonbonded forces. Moreover, F_{mfield} indicates the time spent on the calculation of intermolecular nonbonded forces, and F_{bond} and F_{angle} indicate the time spent on the calculation of forces due to bonds and angles, respectively. At last, in serial MD-SCF program, D_{update} indicates the time spent to update the coarse-grained densities from the particle positions at every 300 time steps.

From Figure 4, it is clear that the computation of nonbonded forces in serial MD simulations is the most expensive part, which takes almost the 80% of total simulation time. The cost of nonbonded forces (100,488s) is about 70 times larger than those of bond (1505s) and angle forces (1378s). On the other hand, in MD-SCF simulations, the expensive calculation of intermolecular nonbonded forces is replaced by the calculation of forces between single particles and the density field. This makes the overall simulation time of the MD-SCF simulation much smaller. As a result, the serial MD simulation takes 127,742s and the corresponding serial MD-SCF simulations takes only 21,407s (about six times fast as serial

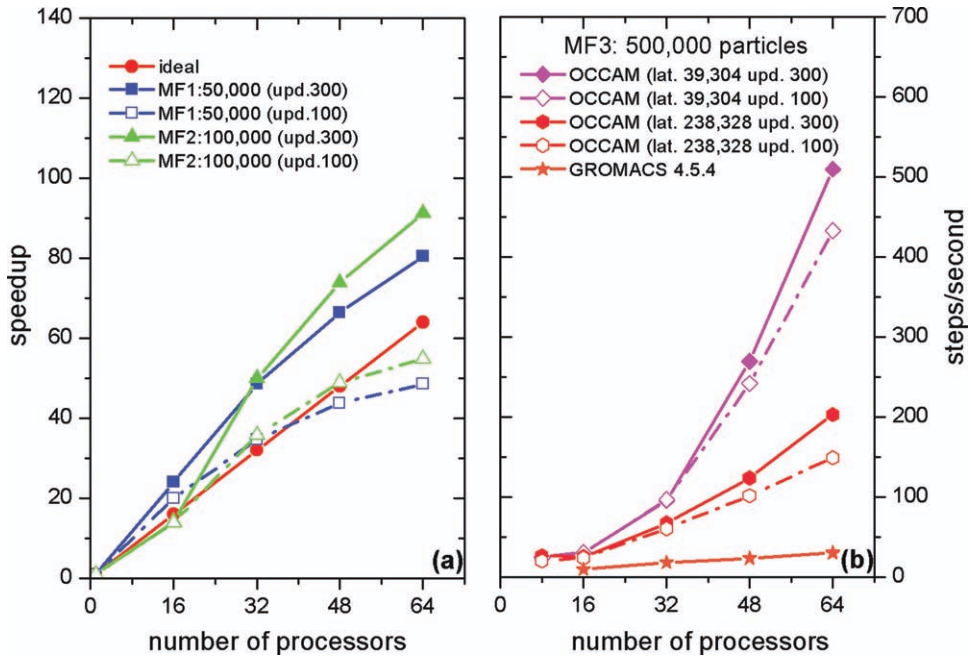

Figure 5. Benchmarks of parallel MD-SCF program with speedup for the monoatomic fluid (MF) systems a) MF1 (blue curves with squares) and MF2 (green curves with triangles). For both systems different update frequencies of 100 (empty symbols and dash dot lines) and 300 time steps (filled symbols and solid lines) have been considered. Ideal speedup (red line with circle). b) Performances of parallel MD-SCF program as steps/s for MF3 system in comparison with GROMACS 4.5.4 (orange curve). Results of OCCAM using 39,304 lattice points (purple curves) and 238,328 lattice points (red curves) are shown. Particle-field MD simulations have been done using update frequency of 100 time steps (empty symbols and dash dot lines) and 300 time steps (filled symbols and solid lines), respectively.

Table 2. Complexity and timing of different operations of parallel MD-SCF program for the system of RC1 including 30,000 particles by using 1 cpu (parallel version) as serial run, 32 cpus, and 64 cpus tested on Brindisi.

Operation	Complexity	Parallelized	1 cpu		32 cpu		64 cpu	
Density	O(Nparticle/P)	Yes	9. 4s	0.04%	0.9s	0.14%	1.6s	0.38%
MPI_ALLREDUCE communication	O(Nlattice)		-	-	24.5s	3.81%	49.6s	11.92%
Density derivatives	O(Nlattice)	No	111.4s	0.48%	110.0s	17.09%	110.0s	26.44%
Fbond	O(Nbonds/P)	Yes	1980.0s	8.50%	55.4s	8.61%	27.5s	6.61%
Fangle	O(Nangles/P)	Yes	2059.9s	8.84%	54.7s	8.50%	27.3s	6.56%
Fnbpairs	O(Nnbpairs/P)	Yes	1531.0	6.57%	29.4s	4.57%	9.3s	2.24%
Fparticle-field	O(Nparticles/P)	Yes	7969.9s	34.21%	86.4s	13.42%	35.3s	8.48%
Andersen Thermostat	O(Nparticles/P)	Yes	6844.3s	29.38%	211.7s	32.89%	106.9s	25.69%
Integrate	O(Nparticles/P)	Yes	1692.4s	7.26%	45.9s	7.13%	16.2s	3.89%
Others			1100.1s	4.72%	24.8s	3.85%	32.5s	7.81%
MDcycle			23398.7s		643.7s		416.1s	
Speedup			1		36.4		56.2	

The operations are timed using TAU profiler version 2.19 with simulations of 1,000,000 time steps. The total time of every operation and contribution as percentage in the total simulation time are reported.

MD simulations). Furthermore, differently from serial MD, the serial MD-SCF simulation is characterized by a flat profile. In the serial MD-SCF simulation, the most expensive parts of the simulation are the interpolation of density and density gradients (occupy 36.97% of total simulation time) and the calculation of the intramolecular interactions (the sum of bonds, angles, and intramolecular nonbonded forces evaluation is about 18% of the total simulation time). In the end, the density-update from the particle positions and the calculation of density derivatives now take a negligible percentage (0.005%) in the total simulation.

Comparison of serial MD-SCF and parallel MD-SCF program profiles

The same simulation system RC1 reported in the previous section has been used to analyze the program performance of the parallel MD-SCF method and to compare the serial and the parallel runs. To do a homogeneous comparison, the performances have been compared using the parallel program running on 1 cpu as serial run and the same parallel profiler (TAU version 2.19) has been used to analyze the program performances. Simulation time corresponding to different parts of parallel MD-SCF program, as well as their percentages of the contributions to the total simulation time is reported in Table 3 for runs using 1, 32, and 64 cpus.

Table 3. Parallel efficiencies of OCCAM and GROMACS 4.5.4 with double precision using 16 cpus, 32 cpus, 48 cpus, and 64 cpus, respectively.

MF3 system (steps/s)						
Program						
	Lattice points	Update frequency	16cpu	32cpu	48cpu	64cpu
OCCAM (double precision)	39,304	300	30.49	95.27	269.28	509.72
	39,304	100	29.98	96.46	241.62	432.51
	238,328	300	24.85	67.12	123.32	202.92
	238,328	100	23.78	60.13	101.49	148.92
GROMACS 4.5.4 (double precision)			9.38	18.01	23.23	29.89

In the case of serial MD-SCF simulation, almost 88% of total simulation time is spent mainly in three parts of the program. As reported in Table 2, the first most expensive operation is the calculation of particle-field forces (its percentage in the total simulation time is 34.21%). This is reasonable because these calculations are performed at every time step and involve interpolations of density derivatives at the position of each particle. Another expensive part is the one related to the thermostat (its percentage in the total simulation time is 29.38%), because in this case the operation is performed at every time step and involves the generation of a random number per particle (a fraction of the particles further takes three random numbers from a Gaussian distribution). Finally, the calculation of intramolecular forces, including bonds, angles, and non bonded force (occupying 8.50%, 8.84%, and 6.57%, respectively) takes about 24% of the total simulation time.

It is interesting to note that, according to the parallelization scheme proposed here, all these expensive operations have been fully parallelized. In fact, the time needed for the calculation of particle-field forces decreases from 7970s for the serial run to 86s (only 1.1% of the time spent in the serial run) and 35s (0.4% of the time spent in the serial run) for the parallel runs on 32 and 64 cpus, respectively. Moreover, similar performances of the parallel program are obtained for the calculations involving the thermostat and the intramolecular forces (see Table 2). As explained before, the calculation of density derivatives is not parallelized. Thus the time spent for this calculation is more or less the same for 1 cpu, 32 cpus, and 64 cpus, which take the values 111.4s (0.48%), 110s (17.09%), and 110s (26.44%), respectively. In the mean time, the cost of the communication between processors in the parallel runs is not large, being only 3.81% and 11.92% of the total simulation time for runs on 32 and 64 cpus, respectively.

Therefore, the overall result is a speedup on the total simulation time (last row of Table 3) close to

the ideal behavior. Speedups of 36.4 and of 56.2 are obtained for simulations running on 32 and 64 cpus, respectively. In the consideration of systems composed of small number of particles (30,000 particles), these speedups show how efficient can be the parallel hybrid MD-SCF simulations. (Please see Fig S1 in the Supporting Information)

Parallel benchmarks

To further assess the performance efficiency of parallel MD-SCF program, the nonsetup time[58] (defined as the wall-clock time that is the sum of three contributions: CPU time, I/O time, and the communication channel delay used to perform MD cycles) is considered. The setup time, including data reading, generating the initial velocities, and setting up the lists of bonds and angles and excluded volume interactions for intramolecular interactions, is not taken into consideration. The speedups[58] (sp) are calculated from eq. (13) and plotted as functions of the number of processors

$$sp = \frac{t_1}{t_p}, \tag{13}$$

where t_1 is the CPU time of a single processor run, and t_p is the CPU time of a run with P processors. Furthermore, to get the costs in terms of real time, the number of steps performed per second (steps/s) and the number of steps performed per day (steps/day) have been considered as a measure of program efficiency.

According to the MD-SCF simulation scheme described before, the two parameters regulating the degree of coarse-graining of the density field are the cell size l and the density-update frequency Δt_{update}. Larger cell sizes lead to more collective density fields. The value of the density-update frequency has to be chosen in a way that the approximation of slow variation of the field with respect to the particle displacement is valid between two consecutive density updates.

From the point of view of the performances of parallel MD-SCF simulations, the cell size l and the density-update frequency Δt_{update} are connected with the amount of the data that have to be communicated among processors and the frequency of the communication operations, respectively. Furthermore, larger cell sizes correspond to the collection of more particles in a single cell and then to a smaller number of cells, which makes the amount of the communication of data lower. Similarly, lower density-update frequencies (corresponding to larger update intervals Δt_{update}) correspond to less communication operations. The density-update frequency is Δt_{update}, and thus the number of communication operations is $t_{sum}/\Delta t_{update}$ (t_{sum} is the total time spent in the MD-SCF simulations).

In this section, simulation results using different cell sizes and different density-update frequencies are discussed and compared for several systems. The values of the cell sizes and the density-update frequencies used for all the simulation tests in the following are chosen in a range in which it is possible to reproduce reference results obtained from full MD simulations for several systems.[30,31,54]

Monoatomic fluid

The first set of benchmarks is the simulations of a monoatomic fluid at equilibrium density of the coarse-grained model of water described by the MARTINI force field.[6] The reason of this choice is that, especially in applications relevant for biological systems, the main part of the computational efforts are devoted to the calculation of the nonbonded forces between water or more solvent molecules.

The speedups for the systems of MF1 and MF2 in Table 2 are calculated on Brindisi with 8 cpus, 16 cpus, 32 cpus, 48 cpus, and 64 cpus, respectively and shown in Figure 5a. In both of these systems including 50,000 and 100,000 particles, the cell size $l = 0.705$ nm and different density-update frequencies of $\Delta t_{update} = 100$ and $\Delta t_{update} = 300$ time steps are considered, respectively. Due to a better balance between computation and communication costs, the system MF2 with a larger number of particles shows better performances starting from 32 cpus. From these results, it is clear how the density-update frequency has an important effect in the performance of parallel MD-SCF simulations. In fact, for both systems, larger speedups are obtained from larger values of Δt_{update}. Furthermore, it is worth noting that for the simulation performed with larger values of Δt_{update}, the speedup is larger than the ideal linear speedup. This behavior called superlinear speedup in the high performance computing literature[59] is usually due to a more efficient access to cache memory running on more processors and also depends on the computer architecture. In our case, according to the parallelization scheme used, the memory allocations involving the density and its derivatives are constant when changing the number of processors. In contrast, the allocations of vectors containing coordinates, bonds, and angles etc. decrease when the number of processor increases. The effect of memory allocation is more pronounced in the tests performed using $\Delta t_{update} = 300$ timesteps in which the number of communication operations is less and the total communication amount of data is lower.

Furthermore, to illustrate the scaling behavior for large scale simulations, a system consisting of 500,000 particles (MF3) has been tested on Brindisi and shown on Figure 5b. As explained before, the cell size l also affects the communication amount of data needed for the description of the coarse-grained density on the lattice. Thus, the system MF3 with two different cell sizes and two different density-update frequencies has been performed and tested. Two tests with a different number of cells per cartesian x, y, and z directions of a cubic box have been chosen as 33 and 61, which correspond to cell sizes $l =$ 1.316 and 0.705 nm, and are in a range available for the reproduction of the reference results of full atomistic MD simulations.[30,31,54] In terms of the number of lattice points, they correspond to $34^3 = 39{,}304$ and $62^3 = 238{,}328$ points shown in Table 1.

In the following the performances of hybrid particle-field simulations are compared with MD simulations using Lennard-Jones pair potentials for nonbonded interactions. The meaning of this comparison is to show with real examples how much is gained by using a field based approach. To compare parallel

hybrid particle-field MD-SCF simulations with parallel particle–particle full MD simulations, the performance of the same system MF3 obtained from efficient program GROMACS 4.5.4 with double precision has been also tested on Brindisi and shown (orange curve) in Figure 5b and Table 3. In GROMACS 4.5.4, it is implemented a domain decomposition scheme by using dynamical load balancing for parallel simulations.[60] From the figure it is clear that, the performance of the parallel MD-SCF simulations with particle decomposition algorithm is better than that of the parallel MD program GROMACS 4.5.4 using domain decomposition scheme, especially when a larger number of processors are used. For example, the efficiency of parallel MD-SCF program OCCAM using 64 cpus with number of lattice points of 39,304 and update frequency of 300 time steps is 16.8 times of that performed with parallel MD program GROMACS 4.5.4 as shown in Table 3.

We stress that we compared two different parallel decomposition algorithms in the benchmarks between parallel performances of classical MD (GROMACS 4.5.4) and hybrid particle-field MD-SCF simulations. To have a more homogeneous comparison, we should compare both simulations with the same parallel decomposition algorithm (i.e. particle decomposition). In the case of GROMACS 4.5.4, as reported in ref. [60], the performance of the parallelized program using a particle decomposition scheme is lower than that with the domain decomposition algorithm especially when a large number of CPUs is used. For instance, for membrane/protein systems (121,449 atoms), the simulation speed performed with parallel particle decomposition scheme is 0.4 times as the one performed using the domain decomposition algorithm with 32 CPUs. Therefore, for a more homogenous comparison between particle–particle and particle-field methods using the same particle decomposition algorithm, even better performances (for example, more than 18.6 times) could be expected for the particle-field MD-SCF method.

Moreover, a larger system MF4 consisting of 1,000,000 particles in Table 1 is also performed and tested by parallel MD-SCF program OCCAM and GROMACS 4.5.4, respectively. From Figure S3 (in the Supporting Information), we can find similar results with MF3 systems, which indicate that the parallel hybrid particle-field MD-SCF simulations are faster than the parallel particle–particle MD simulations especially for a large number of processors. For example, program OCCAM with 64 cpus performs 62.9 steps/s (at $\Delta t_{update} = 300$ time steps and $l = 0.705$ nm), which are 3.4 times as that of GROMACS 4.5.4 (18.7 steps/s on 64 cpu). For this large system, the reasonable efficiency indicates that parallel particle decomposition algorithm is very suitable for the implementation of parallel MD-SCF simulations and helps the simulation of a large system size reach the equilibrium in a reasonable time scale.

Large scale systems with intramolecular interactions

In this section large scale systems (ranging from about 300,000 to 1,200,000 particles) including intramolecular interactions have been considered. In particular, simulation speeds of coarse-grained models of water and lipids, as well as mixtures of rod and coil homopolymers, will be discussed.

Benchmarks of the large scale systems have been performed by running on 64, 80, and 96 cpus using $\Delta t_{update} = 300$ time steps using a cell sizes $l = 1.5\ \sigma$ and $2.5\ \sigma$ (σ is the diameter of one particle). In this range test simulations performed on the systems of lipids and water indicate a good reproduction of density profiles of the reference system calculated with classical MD.[54] For these larger systems the simulations have been performed on a different machine (Cresco 1) that allowed us to run jobs on more than 64 cpus. To reproduce conditions that are representative of real application runs, we performed these test jobs by submitting them using a job scheduler. On this machine, the job scheduler can address the same parallel simulation to a different number of processors on different nodes according to their availability, and then the test results can be different. For this reason we performed three different runs for each test simulation and as a result we report the average performance with a small error bar.

The performances have been measured as million of time steps/day and reported in Figure 6 and Table 4. For the

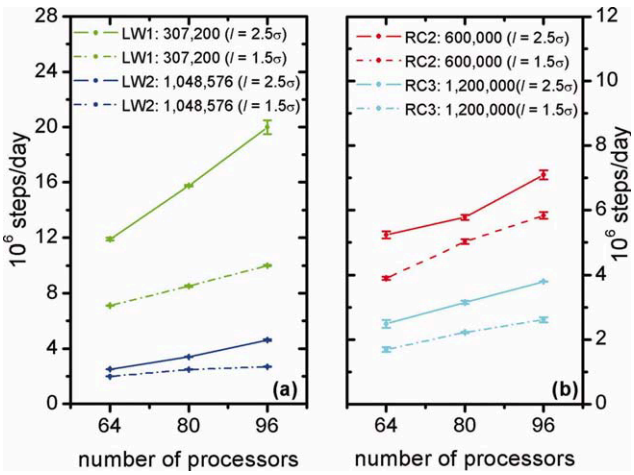

Figure 6. Performances of parallel MD-SCF OCCAM program as 10^6 steps/day for large scale systems tested on Cresco1. a) Lipid and water systems LW1 (green curves) and LW2 (blue curves) and b) rod and coil polymer mixtures RC2 (red curves) and RC3 (cyan curves). Both LW and RC systems have been tested using update frequency of 300 time steps. Two different cell sizes l have been considered in these simulations, including $l = 1.5\sigma$ (empty symbols and dash dot lines) and $l = 2.5\sigma$ (filled symbols and solid lines), respectively. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

systems LW1 (307,200 particles), the cell size shows a very big effect on the efficiency. For example, the simulation speed tested on 96 cpus is about 10 million steps/day using smaller cell size ($l = 1.5\sigma$). However, the efficiency with larger cell size ($l = 2.5\sigma$) can achieve two times faster than this (about 20 million steps/day). Furthermore, in the larger system LW2 (1,048,576 particles), the effect of cell size becomes smaller. For example, the simulation speed is 2.7 million steps/day (on 96 cpus) with smaller cell size ($l = 1.5\sigma$), and the efficiency of 4.6 million steps/day (on 96 cpus) is obtained with larger cell

Table 4. Simulation speed of parallel MD-SCF program tested on Cresco1 using a job scheduler with large scale systems, including systems of LW1, LW2, RC2, and RC3.

	Large scale systems (10^6 steps/day)			
System	Cell size l (σ)	64 cpu	80 cpu	96 cpu
LW1	1.5	7.11 ± 0.02	8.52 ± 0.06	10.01 ± 0.04
LW1	2.5	11.88 ± 0.09	15.75 ± 0.07	19.99 ± 0.5
LW2	1.5	1.980 ± 0.01	2.48 ± 0.01	2.69 ± 0.02
LW2	2.5	2.50 ± 0.01	3.39 ± 0.01	4.62 ± 0.05
RC2	1.5	3.90 ± 0.05	5.02 ± 0.07	5.83 ± 0.1
RC2	2.5	5.23 ± 0.11	5.78 ± 0.07	7.09 ± 0.1
RC3	1.5	1.69 ± 0.07	2.22 ± 0.01	2.62 ± 0.08
RC3	2.5	2.49 ± 0.12	3.15 ± 0.05	3.80 ± 0.01

Three different runs for each simulation are tested and the average performances with a small error bar are reported.

size ($l = 2.5\sigma$). However, it is clear that the simulation speed of both 2.7 and 4.6 million steps/day with 96 cpus performed on the lipid and water system including about one million particles will be very helpful to investigate this interesting biological system in a reasonable time scale.

For the systems of RC2 (600,000 particles) and RC3 (1,200,000 particles) in Table 1, we can find that the number of particles in RC3 is two times as that in RC2. Thus, according to particle decomposition algorithm, the ideal simulation speed of a 600,000 particle system should be two times as that of a system including 1,200,000 particles. However, the results in Table 3 and Figure 6 show that the efficiency of RC2 system is 2.2 times as that in RC3 system using the smaller cell size ($l = 1.5\sigma$) and 96 cpus, which is better than the ideal simulation efficiency. This is reasonable, because the communication amount of data mostly depends on the number of lattice points when update frequency ($\Delta t_{update} = 300$ time steps) and cell size ($l = 1.5\sigma$) are constant, in which the number of lattice points 373,248 in RC3 system is more than two times of that in RC2 system (lattice points 185,193), as shown in Table 1. Furthermore, using the cell size ($l = 2.5\sigma$), the number of lattice points 79,507 in RC3 system is less by a factor 2 than that of RC2 system (lattice points 42,875) as is shown in Table 1, which results in that the efficiency of RC2 is less than two times of that in RC3. For example, using 96 cpus and the cell size ($l = 2.5\sigma$), the simulation speed of RC2 system is 7.1 million steps/day, which is 1.9 times as that in RC3 system (3.8 million steps/day) shown in Table 4. In summary, these high efficiencies (several million steps/day) of large scale systems (million particles) performed by parallel MD-SCF simulations not only indicates that particle decomposition algorithm is very suitable for the parallelization of hybrid MD-SCF method, but also makes it possible to investigate large scale complex system in a reasonable time scale.

Conclusions

The parallel implementation of a recently developed hybrid particle-field MD-SCF scheme for MDs simulations has been described. Due to the peculiar formulation of the method considering particles interacting with density fields, the most computationally expensive part of a hybrid particle-field MD-SCF simulation can be efficiently parallelized using a straightforward particle decomposition algorithm. This approach is easy to implement but usually limited for classical MDs due to the large amount of communications of the necessary data. It can be efficiently exploited in the parallelization of MD-SCF simulations.

Several benchmarks performed on different systems show that the two main parameters (the density-update frequency and the cell size) characterizing the density coarse-graining are very important in regulating the performances of parallel runs. Considerable speedups have been achieved for all considered systems. The results of benchmarks indicate that the proposed particle decomposition parallelization scheme is very suitable and efficient, and opens the way to the simulations of large scale complex systems with reasonable computational costs.

Acknowledgments

G. M., Y. Z., and A. D. N. thank the HPC team of Enea (www.enea.it) for using the ENEA-GRID and the HPC facilities CRESCO (www.cresco.enea.it) in Portici, Italy. G. M. expresses his gratitude to Florian Müller-Plathe (Technical University of Darmstadt) for his encouragement to develop the OCCAM code.

Keywords: molecular dynamics · coarse-graining · parallelization

How to cite this article: Y. Zhao, A. De Nicola, T. Kawakatsu, G. Milano, *J. Comput. Chem.* **2012,** *33,* 868–880. DOI: 10.1002/jcc.22883

Additional Supporting Information may be found in the online version of this article.

[1] M. P. Allen, D. J. Tildesley, Computer Simulation of Liquids; Oxford University Press: Oxford, **1989**.
[2] D. C. Rapaport, The Art of Molecular Dynamics Simulation; Cambridge University Press: New York, **1996**.
[3] D. Frenkel, B. Smit, Academic Press: San Diego, California, **2002**.
[4] G. Milano, F. Müller-Plathe, *J. Phys. Chem. B* **2005**, *109,* 18609.
[5] T. Spyriouni, C. Tzoumanekas, D. Theodorou, F. Muller-Plathe, G. Milano, *Macromolecules* **2007**, *40,* 3876.
[6] L. Monticelli, S. K. Kandasamy, X. Periole, R. G. Larson, D. P. Tieleman, S.-J. Marrink, *J. Chem. Theory. Comput.* **2008**, *4,* 819.
[7] M. Karplus, G. A. Petsko, *Nature* **1990**, *347,* 631.
[8] F. Müller-Plathe, *Chem. Phys. Chem.* **2002**, *3,* 754.
[9] W. F. van Gunsteren, D. Bakowies, R. Baron, I. Chandrasekhar, M. Christen, X. Daura, P. Gee, D. P. Geerke, A. Glattli, P. H. Hunenberger, M. A. Kastenholz, C. Oostenbrink, M. Schenk, D. Trzesniak, N. F. A. van der Vegt, H. B. Yu, *Angew. Chem. Int. Ed.* **2006**, *45,* 4064.
[10] M. Praprotnik, L. Delle Site, K. Kremer, *Annu. Rev. Phys. Chem.* **2008**, *59,* 545.
[11] T. Kawakatsu, Statistical Physics of Polymers; Springer-Verlag: Berlin, **2004**.
[12] G. H. Fredrickson, The Equilibrium Theory of Inhomogeneous Polymers; Clarendon Press: Oxford, **2006**.
[13] M. W. Matsen, M. Schick, *Phys. Rev. Lett.* **1994**, *72,* 2660.
[14] F. Drolet, G. H. Fredrickson, *Phys. Rev. Lett.* **1999**, *83,* 4317.
[15] G. H. Fredrickson, V. Ganesan, F. Drolet, *Macromolecules* **2002**, *35,* 16.
[16] Y. Lauw, F. A. M. Leermakers, M. A. Cohen Stuart, *J. Phys. Chem. B.* **2006**, *110,* 465.

[17] D. Q. Ly, T. Honda, T. Kawakatsu, A. V. Zvelindovsky, *Macromolecules* **2008**, *41,* 4501.
[18] E. Dickinson, V. J. Pinfield, D. S. Horne, F. A. M. Leermakers, *J. Chem. Soc. Faraday Trans.* **1997**, *93,* 1785.
[19] A. C. Balazs, C. Singh, E. Zhulina, *Macromolecules* **1998**, *31,* 8370.
[20] J. R. Roan, T. Kawakatsu, *J. Chem. Phys.* **2002**, *116,* 7283.
[21] J. R. Roan, T. Kawakatsu, *J. Chem. Phys.* **2002**, *116,* 7295.
[22] M. Müller, G. D. Smith, *J. Polym. Sci. Part B: Polym. Phys.* **2005**, *43,* 934.
[23] K. C. Daoulas, M. Müller, M. P. Stoykovich, S. M. Park, Y. J. Papakonstantopoulos, J. J. D. Pablo, P.F. Nealey, H. H. Solak, *Phys. Rev. Lett.* **2006**, *96,* 036104.
[24] H. Kang, F. A. Detcheverry, A. N. Mangham, M. P. Stoykovich, K. C. Daoulas, R. J. Hamers, M. Müller, J. J. D. Pablo, P. F. Nealey, *Phys. Rev. Lett.* **2008**, *100,* 148303.
[25] F. A. Detcheverry, H. Kang, K. C. Daoulas, M. Müller, P. F. Nealey, J. J. D. Pablo, *Macromolecules* **2008**, *41,* 4989.
[26] S. W. Sides, B. J. Kim, E. J. Kramer, G. H. Fredrickson, *Phys. Rev. Lett.* **2006**, *96,* 250601.
[27] G. A. Buxton, J. Y. Lee, A. C. Balazs, *Macromolecules* **2003**, *36,* 9631.
[28] R. B. Thompson, V. V. Ginzburg, M. W. Matsen, A. C. Balazs, *Macromolecules* **2002**, *35,* 1060.
[29] J. Y. Lee, Z. Shou, A. C. Balazs, *Phys. Rev. Lett.* **2003**, *91,* 136103.
[30] G. Milano, T. Kawakatsu, *J. Chem. Phys.* **2009**, *130,* 214106.
[31] G. Milano, T. Kawakatsu, *J. Chem. Phys.* **2010**, *133,* 214102.
[32] L. Dagum, R. Menon, *IEEE. Comput. Sci. Eng.* **1998**, *5,* 46.
[33] M. P. I. Forum, *Int. J. Supercomput. App.* **1994**, *8,* 165.
[34] H. J. C. Berendsen, D. van der Spoel, R. van Drunen, *Comput. Phys. Commun.* **1995**, *91,* 43.
[35] T. R. Forester, W. Smith, DL_POLY User Manual; CCLRC, Daresbury Laboratory: Warrington, UK, **1995**.
[36] J. Stadler, R. Mikulla, H.-R. Trebin, *Int. J. Mod. Phys.* **1997**, *8,* 1131.
[37] A. P. Lyubartsev, A. Laaksonen, *Comput. Phys. Commun.* **2000**, *128,* 565.
[38] R. Couturier, C. Chipot, *Comput. Phys. Commun.* **2000**, *124,* 49.
[39] T. P. Straatsma, M. Philippopoulos, J. A. McCammon, *Comput. Phys. Commun.* **2000**, *128,* 377.
[40] S. Roy, R. Y. Jin, V. Chaudhary, W. L. Hase, *Comput. Phys. Commun.* **2000**, *128,* 210.
[41] R. Trobec, M. Sterk, D. Janezic, *Int. J. Quantum. Chem.* **2001**, *84,* 23.
[42] S. Goedecker, *Comput. Phys. Commun.* **2002**, *148,* 124.
[43] C. Ozdogan, G. Dereli, T. Cagin, *Comput. Phys. Commun.* **2002**, *148,* 188.
[44] K. B. Tarmyshov, F. Muller-Plathe, *J. Chem. Inf. Model* **2005**, *45,* 1943.
[45] J. C. Phillips, R. Braun, W. Wang, J. Gumbart, E. Tajkhorshid, E. Villa, C. Chipot, R. D. Skeel, L. Kale, K. Schulten, *J. Comput. Chem.* **2005**, *26,* 1781.
[46] B. FrantzDale, S. J. Plimpton, *Eng. Comput.* **2010**, *26,* 205.
[47] M. S. Friedrichs, P. Eastman, V. Vaidyanathan, M. Houston, S. Legrand, A. L. Beberg, D. L. Ensign, C. M. Bruns, V. S. Pande, *J. Comput. Chem.* **2009**, *30,* 864.
[48] P. Larsson, B. Hess, E. Lindahl, Wiley Interdisciplinary Reviews: Computational Molecular Science **2011**, 1, 93.
[49] D. Finchama, *Mol. Simul.* **1987**, *1,* 1.
[50] W. Smith, *Comput. Phys. Commun.* **1991**, *62,* 229.
[51] S. Gupta, *Comput. Phys. Commun.* **1992**, *70,* 243.
[52] S. Plimpton, *J. Comput. Phys.* **1995**, *117,* 1.
[53] G. S. Heffelfinger, *Comput. Phys. Commun.* **2000**, *128,* 219.
[54] A. De Nicola, Y. Zhao, T. Kawakatsu, D. Roccatano, G. Milano, *J. Chem. Theory Comput.* **2011**, *7,* 2947.
[55] G. Milano, Available at https://www.molnac.unisa.it/occam/, **2007**.
[56] S. L. Graham, P. B. Kessler, M. K. Mckusick, ACM SIGPLAN Notices, In Proceedings of the 1982 SIGPLAN Symposium on Compiler Construction, New York, NY, USA, **1982**.
[57] S. S. Shende, A. D. Malony, *Int. J. High Perform Comput.* **2006**, *20,* 287.
[58] A. Liwo, S. Oldziej, C. Czaplewski, D. S. Kleinerman, P. Blood, H. A. Scheraga, *J. Chem. Theory Comput.* **2010**, *6,* 890.
[59] C. R. Mechoso, J. D. Farrara, J. A. Spahr, *IEEE Parallel Distr. Technol.* **1994**, *2,* 57.
[60] B. Hess, C. Kutzner, D. V. D. Spoel, E. Lindahl, *J. Chem. Theory Comput.* **2008**, *4,* 435.

Received: 8 July 2011
Revised: 17 October 2011
Accepted: 18 October 2011
Published online on 25 January 2012